Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently Amended) A method comprising:

pushing a datum onto a stack by a first hardware-controlled processing thread in a hardware-based multi-threaded processor to make available the datum for a second hardware-controlled processing thread in the hardware-based multi-threaded processor, the processing thread comprising computer instructions that perform a task; and

popping the datum off the stack by a second hardware-controlled processing thread in the hardware-based multi-threaded processor to use the datum, the second processing thread comprising computer instructions that perform a task.

2. (Original) The method of claim 1 wherein the pushing comprises:

executing a push command on the first processing thread, the push command having at least one argument,

determining a pointer to a current stack datum,

determining a location associated with an argument of the push command,

storing the determined pointer at the determined location,

producing a pointer associated with determined location the pointer to the current stack datum.

3. (Original) The method of claim 2 wherein determining a location comprises:

decoding the push command.

4. (Original) The method of claim 2 wherein determining a location comprises:

storing an argument of the pop command in a location associated with the argument of the push command.

5. (Original) The method of claim 2 wherein said push command is at least one of a processor instruction, and an operating system call.

6. (Original) The method of claim 1 wherein popping comprises:

executing a pop command by the second processing thread,

determining a pointer to a current stack datum,

returning the determined pointer to the second processing thread,

retrieving a pointer to a previous stack datum from a location associated with the pointer to the current stack datum, and

assigning the retrieved pointer the pointer to the current stack datum.

7. (Original) The method of claim 6 wherein the location associated with the pointer to the current stack datum is the location that has an address equal to the value of the pointer to the current stack datum.

8. (Original) The method of claim 6 wherein the location associated with the pointer to the current stack datum is the

location that has an address equal to the sum of an offset and the value of the pointer to the current stack datum.

9. (Original) The method of claim 6 wherein the pop command is at least one of a processor instruction or an operating system call.

10. (Original) The method of claim 1 further comprising:

storing data in a memory buffer that is accessible using a buffer pointer having the datum that is pushed onto the stack.

11. (Original) The method of claim 1 further comprising:

using the popped datum as a buffer pointer to access information stored in a memory buffer.

12. (Original) The method of claim 1 further comprising:

a third processing thread pushing a second datum onto the stack.

13. (Original) The method of claim 1 further comprising:

a third processing thread popping a second datum of the stack.

14. (Previously Presented) A system comprising:

a stack module that stores data by pushing it onto the stack and processing threads can retrieve information by popping the information off the stack,

a first hardware-controlled processing thread in a hardware-based multi-threaded processor, the first hardware-controlled processing thread having a first command set, including at least one command for pushing data onto the stack to make available the data for other hardware-controlled processing threads in the hardware-based multi-threaded processor, and

a second hardware-controlled processing thread in a hardware-based multi-threaded processor, the second hardware-controlled processing thread having a second command set, including at least one command for popping the data off the stack to use the data.

15. (Original) The system of claim 14 wherein the first and second processing threads are executed on a single processing engine.

16. (Original) The system of claim 14 wherein the first and second processing threads are executed on separate processing engines.

17. (Original) The system of claim 16 wherein the separate processing engines are implemented on the same integrated circuit.

18. (Original) The system of claim 14 wherein the stack module and the processing threads are on the same integrated circuit.

19. (Original) The system of claim 14 where the first and second command sets are at least one of a processor instruction set and an operating system instruction set.

20. (Original) The system of claim 14 further comprising a bus interface for communicating between at least one of the processing threads and the stack module.

21. (Currently Amended) A stack module comprising:

control logic that responds to commands from at least two hardware-controlled processing threads in a hardware-based multi-threaded processor, the control logic storing datum on a stack structure in response to a push command to make available the datum for other hardware-controlled processing threads in the hardware-based multi-threaded processor and retrieving datum to use the datum from the stack in response to a pop command.

22. (Original) The stack module of claim 21 further comprising a stack pointer associated with the most recently stored datum on the stack.

23. (Original) The stack module of claim 22 further comprising a memory location associated with a first datum on the stack, the second memory location including:

a pointer associated with a second datum which was stored on the stack prior to said first datum.

24. (Original) The stack module of claim 22 further comprising a second stack pointer associated with the most recently stored datum on a second stack.

25. (Original) The stack module of claim 22 wherein the stack pointer is a register on a processor.

26. (Original) The stack module of claim 23 wherein said memory location includes SRAM memory.

27. (Original) The stack module of claim 21 wherein the commands are processor instructions.

28. (Original) The stack module of claim 21 wherein the commands are operating system instructions.

29. (Currently Amended) An article comprising computer logic tangibly embodied in a computer-readable medium which stores the computer logic, the computer logic comprising:

a stack module configured to store data from a first hardware-controlled processing thread in a hardware-based multi-threaded processor by pushing the data onto a stack to make available the data for other hardware-controlled processing threads in the hardware-based multi-threaded processor and to retrieve the data to use the data for a second hardware-controlled processing in the hardware-based multi-threaded processor thread by popping the data off the stack, the stack module being responsive to a first processing thread command to store data on the stack and a second processing thread command to retrieve data from the stack.

30. (Currently Amended) An article comprising computer-executable instructions tangibly embodied in a computer-readable medium which stores the computer-executable instructions, the computer-executable instructions causing a processor to:

store data from a first hardware-controlled processing thread in a hardware-based multi-threaded processor by executing an instruction to push the data onto the stack to make available

the data for other hardware-controlled processing threads in the hardware-based multi-threaded processor; and

retrieve the data for a second hardware-controlled processing thread in the hardware-based multi-threaded processor by executing an instruction to pop the data from the stack for use by the second thread.